

Section III:

**PROPOSED AMENDMENT UNDER 37 CFR §1.121 to the
DRAWINGS**

No amendments or changes to the Drawings are proposed.

Section IV:
AMENDMENT UNDER 37 CFR §1.121
REMARKS

Rejections under 35 U.S.C. §112, First Paragraph

In the Office Action, Claims 1 - 15 were rejected reasoning:

Examiner in the Office Action:

Claims 1-15 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the written description requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention.

Claim 1 recites: "sub sequent to said step of selection, automatically copying said selected information elements into a transfer buffer, thereby concatenating two or more information elements into said buffer, said **transfer buffer being stored in a memory constructed other than a file in a file system**" There is no mention in the original specification of having the **transfer buffer** constructed **other than a file in a file system**. Thus, the limitation includes subject matter that was not described in the original specification.

Applicant respectfully disagrees. By "transfer buffer constructed other than a file in a file system", Applicant is referring to a transfer buffer such as a "clipboard", as disclosed in the specification (see. paragraphs 0005, 0008, 0009, 0019 - 0020, 0042, 0069, 0076 - 0078, 0081, and 0090). In particular, paragraph 0005 discloses a clipboard as being in "memory", not in a file system, and paragraph 0069 refers to storing copied information into memory such as a buffer or clipboard.

Further, the use of the term "clipboard" in the industry refers to a buffer stored in memory, not a file in a file system. Thus, a "clipboard" inherently is not a memory structure in a file system. The cited reference discloses a file system, not a clipboard, so the language was added to the claim in accordance with the specification.

Applicant hereby amends claims 1, 6 and 11 to specify that the claimed transfer buffer "comprises a clipboard in memory", which is fully supported, inherently and literally, by the specification. Withdrawal of these rejections is respectfully requested.

Rejections under 35 U.S.C. §103(a)

In the Office Action, Claims 1, 5 - 6, 10-11 and 15 were rejected under 35 U.S.C. 103(a) as being unpatentable over US patent 6,807,668 B2 to Stern, et al., hereinafter "Stern", in view of non-patent literature "Breaking the copy/paste cycle: the stretchable selection tool" by Apperley et al. (hereinafter "Apperley").

Skill Level in the Art not Established. With respect to the determination of obviousness under 35 U.S.C. 103(a), the Examiner is a fact finder required to resolve *Graham* inquiries ("Examination Guidelines for Determining Obviousness Under 35 U.S.C. 103 in View of the Supreme Court Decision in *KSR International Co. v. Teleflex Inc.*," Fed. Reg., Vol. 72, No. 195, October 10, 2007).

According to the Court in *Graham v. John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), it is critical to determining obviousness under 35 U.S.C. §103 to ascertain the level of ordinary skill in the art, whereas this is pivotal in the language and standard set forth in the law at §103.

In the Office Action, the Applicant has not been notified what was considered to be the level of ordinary skill in the art. It is not clear if any of the criteria to be considered in determining the level of ordinary skill in the art under the third factual inquiry of *Graham v. John Deere*, as set forth in *Environmental Designs, Ltd. v. Union Oil*, 713 F.2d 693, 696, 218 USPQ 865, 868 (Fed. Cir. 1983) and in *Bausch & Lomb, Inc. v. Barnes-Hind/Hydrocurve, Inc.*, 796 F.2d 443, 449-450, 230 USPQ 416, 420 (Fed. Cir. 1986) such as (1) the educational level of the inventor, (2) the type of problems encountered in the art, (3) the prior art solutions to those problems, (4) the rapidity with which innovations are made by others, not including the inventor, (5) the sophistication of the technology, and (6) the educational level of active workers in the field not including the inventor, were considered, and if so, how.

Applicant respectfully submits that a holding of obviousness is improper without this factual determination by the Examiner, and requests withdrawal of the rejections of claims 1 -

15.

Claims 1, 6, and 11. With respect to Claims 1, 6, and 11, it was reasoned:

Examiner in the Office Action:

. . . Subsequent to said step of selection, automatically copying said selected information elements into a transfer buffer, thereby concatenating two or more information elements into said buffer, said transfer buffer being stored in a memory construct other than a file in a file system (sec.2, par.4, "paste buffer"; sec.3.1, par.3; sec.3.3, par. 7; multiple pieces of data are selected and. stored in a system clipboard buffer);

Applicant respectfully disagrees. Applicant submits that Apperley fails to teach "concatenating two or more information elements into said buffer."

By "concatenate", Applicant means a process in which information which is newly "copied" is appended to the contents or information already stored in the clipboard, so that the clipboard only contains one information item at a time, but that information item represents the "collected up" information from multiple copy operations. This is consistent with the term as used in the specification (see paragraphs 0042, 0069, 0074, 0078 - 0079, and 0082), and it is consistent with the extrinsic definition of the term:

Dictionary:

concatenate -verb

1. to link together; unite in a series or chain.

(Source: Dictionary.com Unabridged (v 1.1). Retrieved October 16, 2007,

from Dictionary.com website: <http://dictionary.reference.com>

/browse/concatenate)

Referring to Apperley, section 2, paragraph 4, Applicant respectfully submits that Apperley teaches that GNU Emacs editor keeps a *history list* of items which have been copied, but does not actually "concatenate" those items for the user (e.g. the user must make multiple selections from the list and then operate the paste command to achieve concatenation) (emphasis added by Applicant):

Apperley:

A copy and paste system that comes close to meeting our needs is that provided by the GNU Emacs text editor [8].

The paste buffer holds a **history of copied text selections**.

The 'select and paste' menu option **displays a sub-menu**

of the most recent entries. Any can be pasted into the document. This makes it possible to copy a series of fragments and then perform a series of pastes in some different document or area of the same document, without the need for focus flipping. Because of the limited space available when displaying the sub-menu however, only a tiny amount of text can be displayed, and it can be difficult to recognize fragments. Further, Emacs stores all deleted text in the paste buffer. There is no distinction made between fragments deliberately cut or copied, and text that is simply being removed. As a result the paste list tends to be quite long, and somewhat unpredictable.

Text removed while correcting spelling errors will be entered alongside deliberately copied segments.

Referring to Apperley, section 3.1, paragraph 3, Applicant respectfully submits that Apperley is silent regarding concatenating information in the clipboard, but instead is referring to enabling and disabling their "multiple paste" feature:

Apperley:

In the trial implementation the SST can be switched on and off using the tick box at the lower right of the dialog box (Figure 2). When the SST is active, the pipe and prompt serve to remind the user that they should be trying to find a particular piece of data. In effect, activating the SST constitutes the issue of a multiple paste command. It is appropriate to issue the paste before finding the data, because that is the natural order in which the user addresses the problem.

Apperley's multiple paste feature does not concatenate, but instead copies each item to a

destination immediately after it is selected. For example, refer to Apperley's multiple paste buffers (e.g. more than one buffer, not concatenating in a single buffer) at section 2, paragraph 4. Apperley's disclosure does not contain any other instances of the term "multiple paste".

Referring to Apperley, section 3.3, paragraph 7, Applicant respectfully submits that Apperley is silent regarding concatenating information from multiple copy commands into the single clipboard buffer, but instead is discussing implementation strategies, and even discouraging the assumption that all copy commands are to be treated as commands to transfer information (emphasis added by Applicant):

Apperley:

There were many options for implementing the 'copy and paste' command. The application we have experimented with most often, Internet Explorer, has a COM interface through which selected data might be extracted. In fact, we have chosen to simulate a 'copy' menu selection on the grounds that the event used is common to a large number of applications. The SST software then takes the clipped text from the system clipboard and puts it into the dialog box field. There are also a number of alternative ways in which we might have implemented the 'fire' command. One possibility would have been to simply allow the user to clip using the normal command of the source application and monitor the system clipboard for incoming data. **We rejected this option for two reasons.** One was that applications are not uniform in providing short cut keys or mouse gestures for cutting and we didn't want the user to have to use a menu, with its attendant large mouse movements. **The other, and more important reason was that we couldn't assume that all copy events were intended to deliver data to our system. The user might, for example, want to copy and paste a URL as part of their searching process.**

Such an assumption made by Apperley would lean away from concatenation, which by its nature requires that multiple copy commands be assumed to be a "copy and append" operation

to the contents of the clipboard

In view of Apperley's dissuasion from operational assumption that all copy commands were to be handled as information transfer (and concatenation) commands, Applicant respectfully submits that it would have been unreasonable for one skilled in the art to set aside this portion of Apperley's disclosure, and to proceed to make the combination, substitutions, and modifications as proposed in the rationale for the rejections.

For these reasons, Applicant respectfully requests allowance of claims 1, 6, and 11.

Claims 5, 10 and 15. In the Office Action, Claims 5, 10, and 15 were rejected over Stern in view of Apperley. Since claims 5, 10, and 15 depend from Claims 1, 6, and 11, respectively, and because Stern in view of Apperley fails to teach concatenation of information in a clipboard, Applicant respectfully requests allowance of these claims.

Claims 2 - 4, 7-9 and 12-14. In the Office Action, these claims were rejected under 35 U.S.C. 103(a) as being unpatentable over Stern and Apperley in further view of Tomm et al. (Tomm, US 6,560,608 BI), and in further view of Tsuji et al. (Tsuji, US 5,586,025). Tomm was relied upon for its teaching related to selection of rules to process the copied information, and Tsuji was relied upon for its teaching related to rule management user interfaces.

However, it was not established whether Tomm or Tsuji teach concatenation of selected items in a clipboard buffer. Applicant has word searched these disclosures, and finds no instances of the term "concatenate".

Applicant respectfully submits that Stern in view of Apperley in further view of Tomm in further view of Tsuji (a) fails to teach all of the claimed elements, steps, and limitations, (b) it would not have been obvious to modify so many references as proposed, and (c) it would not have been obvious to modify Apperley against only a selected portion of Apperley's disclosure.

For these reasons, Applicant respectfully requests allowance of claims 2 - 4, 7-9 and 12 - 14.

Respectfully,

/ Robert Frantz /

Robert H. Frantz, Reg. No. 42,553
Agent for Applicants
Tel: (405) 812-5613
Franklin Gray Patents, LLC